Object Tracking using Multiple Motion Modalities

Simon Denman, Clinton Fookes, Sridha Sridharan, and Vinod Chandran
Image and Video Research Laboratory
Queensland University of Technology
GPO Box 2434, Brisbane 4001, Australia
{s.denman, c.fookes, s.sridharan, v.chandran}@qut.edu.au

Abstract

This paper presents an object tracking system that utilises a hybrid multi-layer motion segmentation and optical flow algorithm. While many tracking systems seek to combine multiple modalities such as motion and depth or multiple inputs within a fusion system to improve tracking robustness, current systems have avoided the combination of motion and optical flow. This combination allows the use of multiple modes within the object detection stage. Consequently, different categories of objects, within motion or stationary, can be effectively detected utilising either optical flow, static foreground or active foreground information. The proposed system is evaluated using the ETISEO database and evaluation metrics and compared to a baseline system utilising a single mode foreground segmentation technique. Results demonstrate a significant improvement in tracking results can be made through the incorporation of the additional motion information.

1 Introduction

When performing object tracking, a technique that allows motion to be detected is a common starting point among many tracking systems. A variety of techniques that can detect motion are available, including: foreground segmentation techniques [16, 2]; multi-layer foreground segmentation techniques that are able to split foreground into stationary and moving regions [5, 9]; and optical flow techniques that determine the movement at each pixel from frame to frame [11, 8, 1].

Object tracking systems such as [19, 7, 10] use motion detection as a first step in tracking. Once objects have been located, a variety of methods can be used to maintain the tracking of an object, such as predicting the next position of the object [19, 7], or using the objects colour with histogram matching or colour clustering techniques [10]. Typically, motion segmentation techniques that allow for multi-modal backgrounds are used [16, 2], however, these techniques only detect a single state of motion (i.e. in motion, not in motion).

Another common approach is to use optical flow as a basis for tracking. Yamane et al. [17] proposed a method using optical flow and uniform brightness regions (a section where the optical flow cannot be detected) to track people. Okada et al. [14] uses optical flow and depth information for tracking. These systems [17, 14] rely on averaging the flow for the located object and searching for a region of similar flow vectors in the next frame.

Whilst some tracking systems have sought to combine modalities such as motion and depth [3, 18], or combine multiple inputs in a fusion system [13, 6], systems have avoided combining motion and optical flow. This can possibly be attributed to the computational cost involved in computing both operations every frame.

In this paper a tracking system that utilises a hybrid multi-layer motion segmentation/optical flow [5, 4] algorithm is proposed. The use of such an algorithm as the basis of the system allows multiple modes to be used for object detection. Objects that are in motion, and can have the velocity effectivley estimated can be detected using optical flow. Objects that have stopped moving, can be detected using the static foreground information. All other objects can be detected using the active foreground image, as they would be in tracking systems that use a single motion input. The proposed system is evaluated using a subset of the ETISEO database [12], and compared to a baseline tracking system using a single mode foreground segmentation technique [2] as its basis. Significant improvement as a result of the additional motion information is shown.

This paper is outlined as follows. Section 2 presents

978-1-4244-4242-3/08/$25.00 ©2008 IEEE

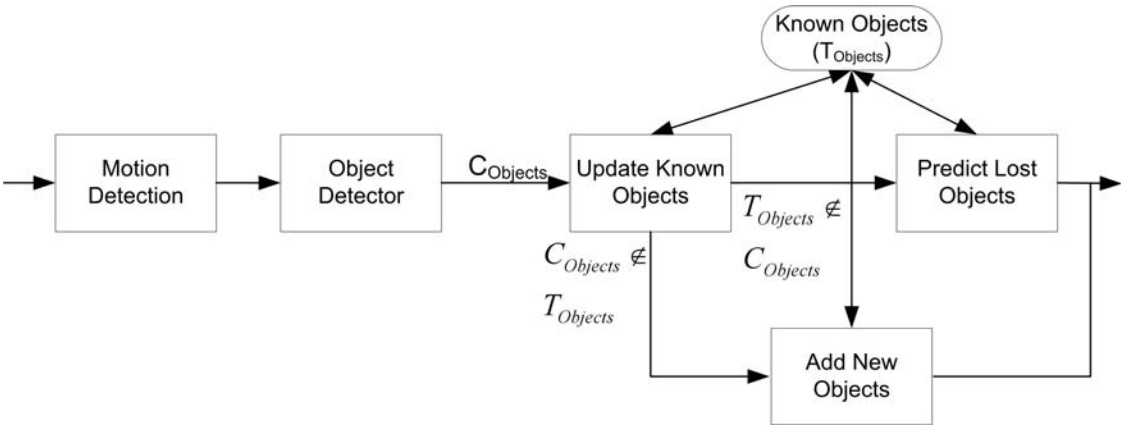


Figure 1. Tracking Algorithm Flowchart.

the baseline object tracking system utilising a single mode foreground segmentation technique. Section 3 presents the proposed hybrid multi-layer motion segmentation and optical flow algorithm. Experimental results are provided in Section 4 while the paper is concluded in Section 5.

2 Baseline Tracking System

The tracking algorithm used in this work is a top-down system (see Figure 1). Motion detection is used to perform initial segmentation, and the resultant motion mask is used by one or more object detectors (possibly in combination with the input image) to detect the target objects. The resulting list of candidate objects, $C_{Obj}(t)$, is compared to the list of tracked objects, $T_{Obj}(t)$. Candidate objects are compared to tracked objects to determine the quality of matches using a fit function, F, which returns a value in the range of 0 to 1. A fit of 1 indicates a perfect match, and a fit of 0 indicates no match. The candidate and track pair which yield the highest fit score are matched, followed by the next lowest until all candidate-track pairs that have a valid match (determined by a threshold on the fit scores) are paired. Any remaining candidates are added as new objects, and any unmatched tracked objects are updated via prediction.

All tracks have a state associated with them and several counters which define how the system handles the track. There are five possible states within the system:

1. Preliminary - Entered into when a track is first created. Tracks in this state must be continually detected.
2. Transfered - Tracks that are moved from another view are created in the transfered state. This is similar to the Preliminary state, but allows for more flexibility when detecting and matching the object.
3. Active - The track has been observed for several frames. Tracks spend most of their time in this state. It indicates that the track has been located in the last frame and its position is known.
4. Occluded - Indicates that the track has not been located in the last frame, either due to occlusion or system error.
5. Dead - The track is to be removed from the system. Tracks in this state are deleted when the current frame's processing ends.

The state transitions are shows in Figure 2, and the transition conditions are outlined in Table 1.

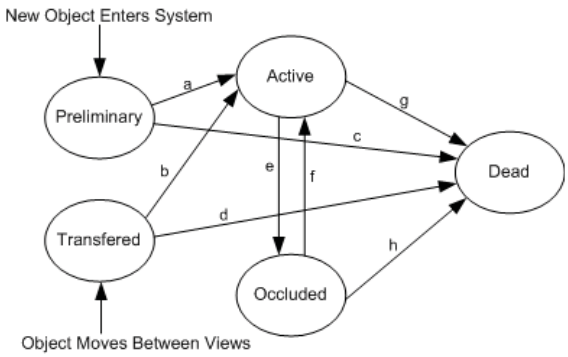


Figure 2. State Diagram for a Tracked Object.

a	Prelim→Active	The tracked object is detected and matched for n_{active} successive frames.
b	Transfered→Active	The tracked object is detected and matched once.
c	Prelim→Dead	The tracked object is not detected and matched for a single frame.
d	Transfered→Dead	The tracked object is not detected and matched for a single frame.
e	Active→Occluded	The tracked object is not detected and matched for a single frame.
f	Occluded→Active	The tracked object is detected and matched for a single frame.
g	Acrive→Dead	The tracked object is explicitly deleted by the system.
h	Occluded→Dead	The tracked object is not detected and matched for $n_{occluded}$ consecutive frames.

Table 1. Transition Conditions.

The system uses the motion detection system proposed by Butler et al. [2], and the output of this is used to detect people and vehicles.

Person detection is performed by splitting the image into sub-regions which contain concentrated areas of motion, and then locating heads and fitting ellipses within each region [7, 19]. Working within subregions overcomes problems caused by people occupying a common column of the image causing inaccurate vertical projections. Heads are detected by combining the vertical projection and pixel height of the top contour (to aid in overcoming problems caused by holes in the motion image), and finding local maxima; which are then filtered by analysing the surrounding region. Ellipses are fitted to the valid heads at an aspect dependent on the candidate head, and if there is a suitable occupancy (motion within the bounds of the ellipse) the candidate is accepted. This process is used for the detection of new tracks, and to support the condensation filter tracking. The optical flow results are used to aid both the motion based detection routines and the condensation filter.

Vehicles are detected by locating large areas of motion, where there is a high concentration of motion pixels in the regions bounding box (i.e. most pixels are in motion), as most vehicles are roughly rectangular in shape. The detection process runs in two stages, the first simply groups large regions of motion together to form a list of initial vehicle candidates. The second analyses this initial list further, checking for overlapping objects to create a list of final vehicle candidates. This final list is then used by the system to update existing tracks and create new tracks.

3 Proposed Tracking System

A tracking system is proposed that utilises a motion segmentation routine capable of segmenting the foreground into regions that are moving and regions that have stopped (active and static) and is able to simultaneously calculate optical flow [4, 5]. The use of such a segmentation algorithm allows detection to be performed using the three available modes of motion, depending on the object characteristics. The motion algorithm used also provides a feedback mechanism, which allows an external system (i.e. a tracking system) to alter the weights of the background model to ensure that objects of interest are not incorporated into the background.

Objects that are in motion, and can have the velocity effectively estimated can be detected using optical flow. Objects that have stopped moving, can be detected using the static foreground information. All other objects can be detected using the active foreground image, as they otherwise would be.

3.1 Detecting Using Optical Flow

Optical flow can be used to aid in the detection of objects that are currently moving in the scene. If an object has been observed for several frames, a reasonable estimate of its velocity can be made. This velocity can be used to extract a candidate region based on matching optical flow values, using the equation,

$$C(n,t) = |U(t) - n_u| < T_u \& |V(t) - n_v| < T_v, \quad (1)$$

where $C(n,t)$ is the candidate image based on the optical flow information for the tracked object n at the time t, U and V are the horizontal and vertical flow images, n_u and n_v are the expected horizontal and vertical velocities for the tracked object n, and T_u and T_v are the optical flow error tolerances for the object detection.

The operation is applied over a region that corresponds to the expected position of the target object. This position is based on the previous observed position, offset by the expected movement. The region is padded (expanded) by a few pixels, (the exact amount varies depending on the dataset, size and speed of objects, and the frame rate), to account for errors in the previous detection, or changes in direction.

The extracted region is likely to be incomplete (i.e. the region may contain holes), due to inconsistencies in the optical flow. To counter this, a morphological close operation is performed on the region. This region is then processed in the same manner as for regular object detection, except any detected candidate can only be matched to the intended target track, n. Like

the previously discussed object detection procedures, any detected and matched region is removed from the motion detection images to prevent the same motion being assigned to multiple objects.

This detection process can only be applied to objects in the *Active* state. This ensures that the object has been tracked for a suitable number of frames to allow a reasonable estimate of the velocity to be made.

3.2 Detecting Using Static Foreground

When an object stops, the optical flow for the object (or at least a significant portion of the object) becomes zero, and is therefore not an ideal mode for detection (the background, as well as any number of other stopped objects will also have an optical flow of zero). The static foreground output from the proposed motion detector, and colour can be used to detect objects in instances such as these.

To allow the system to effectively manage the tracking of moving and stationary objects, the *Active* state is divided into two sub-states, *Moving* and *Static*. Figure 3 shows the updated state diagram that incorporates the two types of tracked objects (moving and static). Objects can only enter and exit the *Active* state as moving objects. *Static* objects can only occur when an object that has been observed comes to a stop (i.e. an object cannot suddenly appear in the image and then not move), and cannot suddenly disappear either, (if detection of a static object fails, it is assumed that it is due to the object moving, so it is no longer static). Objects enter the *static* state when the average velocity calculated according to bounding box position drops below T_{static}.

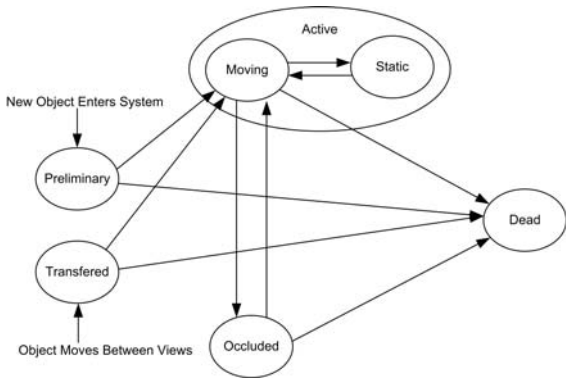


Figure 3. State Diagram Incorporating Static Objects.

To detect objects in the static foreground image, a method to determine when an object becomes stationary is required. This can be achieved by monitoring the objects average velocity over several frames, (the average velocity is defined as the average movement of the centre of the objects bounding box). When this velocity approaches zero (less than a threshold T_{static}), the object is considered stationary and the tracked object enters the *Static* sub-state. Detection of this track is now possible using the static foreground image. However, it may take several more frames for static pixels to appear in the motion images depending on the threshold for static pixels in the motion detector. Optical flow is not used to ascertain if an object is stationary, as a stationary object does not necessarily have an average optical flow of zero. For example, a person might be standing still but waving their arms, which will yield only a small (if any) change in the bounding box (and average velocity calculated based on position), but result in non-zero optical flow.

When a tracked object, $T_{obj}(n)$, enters the *static* sub-state, a template image is created, $I_{ST,n}$. This template is set to a size equal to the width and height of $T_{obj}(n)$, plus a small tolerance to account for any detection and segmentation error in recent frames, which is typically no more than 3 pixels. The template image indicates what pixels belong to $T_{obj}(n)$ and their respective motion mode (static foreground and the layer, active foreground, or the pixel does not belong to $T_{obj}(n)$). A static tracked object may consist of some active pixels, (i.e. a person may be standing still except for their head), and some pixels may change state from active to static and vice versa whilst the object is static, (i.e. a person may be standing still, move an arm, and then be still again).

$I_{ST,n}$ does not store colour as this would be redundant. Assuming that a static pixel remains present at a given location, the colour for that pixel is unchanging, whilst it can be assumed that any active pixels are likely to have a changing colour. When a new static pixel is added to $I_{ST,n}$, its colour is checked against the histogram of $T_{obj}(n)$ to check if that colour belongs to $T_{obj}(n)$, and is only accepted if the colour is present. For any active pixels that are preset, their colour is verified each frame, as there is no way of knowing that the active pixel present at x, y at time t, is the same pixel at time $t + 1$.

$I_{ST,n}$ is used to detect the object in subsequent frames after its creation. For each pixel in $I_{ST,n}$, the algorithm checks if the state indicated by the template is still valid. For example, if the template indicates a static pixel in layer 1, the algorithm checks if a static pixel at layer 1 is present within the static foreground image; if so, this pixel has been detected and verified. If

the expected state cannot be detected, then additional states are checked, (i.e. check the active foreground). Once $T_{obj}(n)$ has been flagged as stationary, and $I_{ST,n}$ has been created, $I_{ST,n}$ cannot be resized, (i.e. the detected object will remain the same size while stationary).

Figure 4 shows an example of the detection and update process using the template image. In Figure 4, the input template contains pixels in both the static foreground state (blue) and active foreground state (green). For pixels in the initial template, the system checks if the mode indicated is still valid, and if so, that state remains. For pixels in the template where the state no longer exists (such as the static pixels in the upper middle of the template), the system checks for other motion modes which may be valid. The resultant updated object template is then stored and used in the next frame.

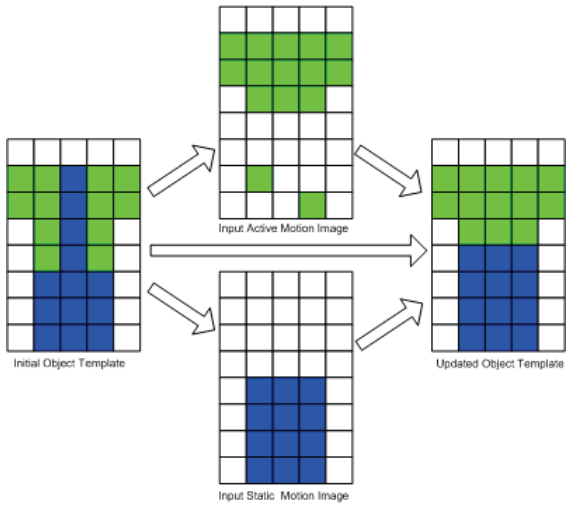


Figure 4. Static Object Detection using the Template Image.

Only the template image is used for detection until the object begins to move again. Movement can be detected by a significant increase in the amount of active foreground present in the template image, or by a decrease in the number of pixels detected as being part of the objects. Movement may also result in failure to detect the object depending on scene characteristics; it is expected that for a static object, it should be detected every frame, and a failure to detect indicates that the object is no longer static. In the case of the object detection failing, the object immediately ceases to be static and object detection is reattempted using the other detection methods. If movement is detected either by a decrease in the number of pixels belonging to the object, or a large decrease in the number of static pixels, then in the next frame the system will revert to other detection routines to locate the object.

3.3 Integration into the Tracking System

The additions to the detection system are incorporated into the system as shown in Figure 5.

Known objects, (those that have been detected in the last frame), are detected and updated first, using the methods described in Sections 3.1 and 3.2, depending on if the object is stationary. This process is shown in Figure 6. For known objects that are successfully detected, their motion is removed from the motion images. The adjusted motion images are then processed by the object detection routines to locate any remaining objects in the scene. At the end of each frame, the locations of the known objects are used to provide feedback to the motion detector, to ensure that motion that has been associated with an object remains separate from the background. Motion that is not associated with an object will gradually be incorporated into the background.

The process that detects known objects is shown in Figure 6. For an object to be detected, it must be in either the *Active* or *Occluded* states. Objects are required to be in the system for several frames prior to this detection to allow an estimate of the optical flow to be made for detection. Static detection will not occur this quickly as the time required for a pixel to be considered stationary far exceeds the time required for an object to be considered *Active* or *Occluded*. Both processes produce one or more candidate objects which are matched to the intended target in the same manner as the original system. If a match is found, the target is updated. If not, the system attempts to detect the object using the standard detection methods once all other known objects have been processed.

4 Results

The proposed tracking system and its improvements are evaluated using a subset of the ETISEO database [12] and the ETISEO evaluation tool [1]. The ETISEO evaluation was run in 2006 to evaluate tracking and event recognition systems.

Tracking output is compared to the ground truth data using the ETISEO evaluation tool. The ETISEO

[1] ETISEO resources such as the database and evaluation tool can be downloaded at http://www-sop.inria.fr/orion/ETISEO/index.htm

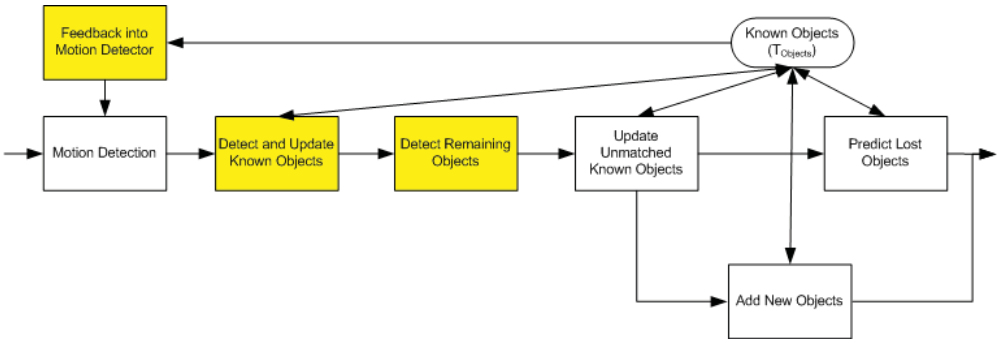


Figure 5. Tracking Algorithm Flowchart with Modified Object Detection Routines (additions/changes shown in yellow).

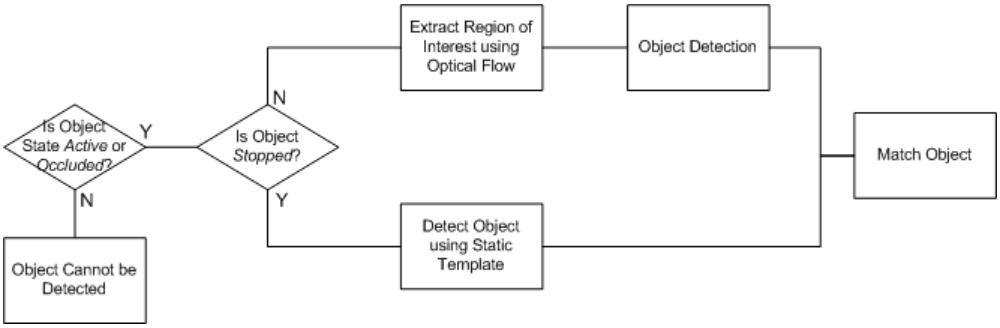


Figure 6. Process for Detecting a Known Object.

evaluation defined several metrics for gauging the performance of tracking systems, which are split into five groups:

1. Detection
2. Localisation
3. Tracking
4. Classification
5. Event Recognition

Results for the proposed tracking systems will be evaluated using metrics from the first three groups, (as there is no classification or event recognition tested with the proposed system). Each group of metrics contain several metrics to evaluate specific areas of interest and a global metric, which is the average of the all metrics within the group. Our evaluation will simply use the overall metrics for detection, localisation and tracking to evaluate the system performance. All metrics result in a value in the range $[0, 1]$, with 1 being a perfect result, and 0 being complete failure. Detailed information on how the metrics are formulated can be found in [15].

The proposed system is evaluated using the RD6 and RD7 datasets from set two of the ETISEO database [12]. These datasets show sequences of a public roadway, with a large number of people and vehicles. Several vehicles stop temporarily, either to let people out or to park. Quantitative results are shown in Table 2 while example visual tracking results are illustrated in Figures 7 and 8. Comparisons are made to the baseline system (see Section 2) to demonstrate the improvement that can be achieved by using multiple modes of motion in combination. Comparisons are not (and can not) be made to other multi-modal systems (such as those that combine motion and stereo [3, 18], or visual and thermal [13, 6]) due to a lack of available data.

The performance of the RD datasets (see Table 2) is significantly better when using the modified system incorporating the new motion detection and detection routines. Significant increases in both the overall detection and tracking metrics are observed. This is a result of the modified systems ability to continue to

System	Data Set	Overall Detection	Overall Localisation	Overall Tracking
Baseline	RD6	0.75	0.95	0.48
Baseline	RD7	0.59	0.92	0.39
Baseline	**Average**	**0.67**	**0.93**	**0.44**
Proposed	RD6	0.74	0.94	0.58
Proposed	RD7	0.60	0.93	0.47
Proposed	**Average**	**0.67**	**0.94**	**0.53**

Table 2. RD Dataset Results using ETISEO Evaluation Metrics.

track objects once they have been stopped for a period of time through the use of multi-layer motion detection and feedback to prevent the motion from being incorporated into the background. An example of this is shown in Figure 7. The top row shows the output of the baseline system while the bottom line shows the output of the tracking system with the proposed modifications.

The loss of tracking on stationary objects due to them becoming part of the background also results in invalid tracks when the objects begin to move again. This results in objects being detected in the place where the car was, as the motion detector has wrongly learned that the primary background mode for that region is the car.

The use of static foreground and a detection routine to locate objects that have stopped, and are visible in static foreground, also results in improved detection and tracking when the objects begin to move again. Figure 8 shows a situation where a car that has been parked begins to move again, (on the far side of the road). The baseline system is able to detect a car, but is unable to correctly localise it due to the errors in motion detection caused by the car being incorporated into the background. As a result, the car is not tracked correctly and an object is falsely detected at the location where the car was, and this results in further tracking errors when a second car passes later on. The improved tracking system using the proposed motion detection does not suffer from these problems, as the parked car is never moved into the background, and so there is no false motion when it begins to move again.

5 Conclusion

This paper has described how a hybrid multi-layer motion segmentation and optical flow algorithm can be used to improve tracking performance within an object tracking system. Tracking experiments conducted on the ETISEO database and using the ETISEO evaluation metrics demonstrated the ability of the hybrid multi-layer algorithm to improve tracking results significantly over those generated with a system utilising a single mode foreground segmentation technique. The incorporation of this additional motion information allows multiple modes to be used for object detection. Objects that are in motion, and can have the velocity effectively estimated can be detected using optical flow. Objects that have stopped moving, can be detected using the static foreground information. All other objects can be detected using the active foreground image, as they would be in tracking systems that use a single motion input. In particular, the proposed system significantly improves tracking results due to the ability to continue to track objects once they have been stopped for a period of time through the use of multi-layer motion detection and feedback to prevent motion being absorbed into the background. The algorithm also does not suffer heavy localisation errors due to false motion when objects are incorrectly incorporated into the background. Future work will investigate the incorporation of multiple motion modalities within an event management framework associated with the object tracking system.

Acknowledgments

This project was supported by the Australian Government Department of the Prime Minister and Cabinet.

References

[1] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Fourth International Conference on Computer Vision*, pages 231 – 236, 1993.

[2] D. Butler, S. Sridharan, and V. M. Bove Jr. Real-time adaptive background segmentation. In *ICASSP '03*, 2003.

[3] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, 2000.

[4] S. Denman, V. Chandran, and S. Sridharan. An adaptive optical flow technique for person tracking systems. *Elsivier Pattern Recognition Letters*, 28(10):1232–1239, 2007.

[5] S. Denman, V. Chandran, and S. Sridharan. Robust multi-layer foreground segmenation for surviellance applications. In *IAPR Conference on Machine Vision Applications*, volume 1, pages 496–499, The University of Tokyo, Japan, 2007.

[6] J. Han and B. Bhanu. Fusion of color and infrared video for moving human detection. *Pattern Recognition*, 40(6):1771–1784, 2007.

[7] I. Haritaoglu, D. Harwood, and L. Davis. An appearance-based body model for multiple people

tracking. In *15th International Conference on Pattern Recognition*, volume 4, pages 184–187, Barcelona, Spain, 2000.

[8] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[9] K. Kim, D. Harwood, and L. S. Davis. Background updating for visual surveillance. In *ISVC*, pages 337–346, 2005.

[10] W. Lu and Y.-P. Tan. A color histogram based people tracking system. In *2001 IEEE International Symposium on Circuits and Systems*, volume 2, pages 137 – 140, 2001.

[11] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[12] A. T. Nghiem, F. Bremond, and M. T. V. Valentin. Etiseo, performance evaluation for video surveillance systems. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 476–481, London, UK, 2007.

[13] C. O'Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton. Comparison of fusion methods for thermo-visual surveillance tracking. In *9th International Conference on Information Fusion (ICIF)*, pages 1–7, 2006.

[14] R. Okada, Y. Shirai, and J. Miura. Tracking a person with 3-d motion by integrating optical flow and depth. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 336–341, 2000.

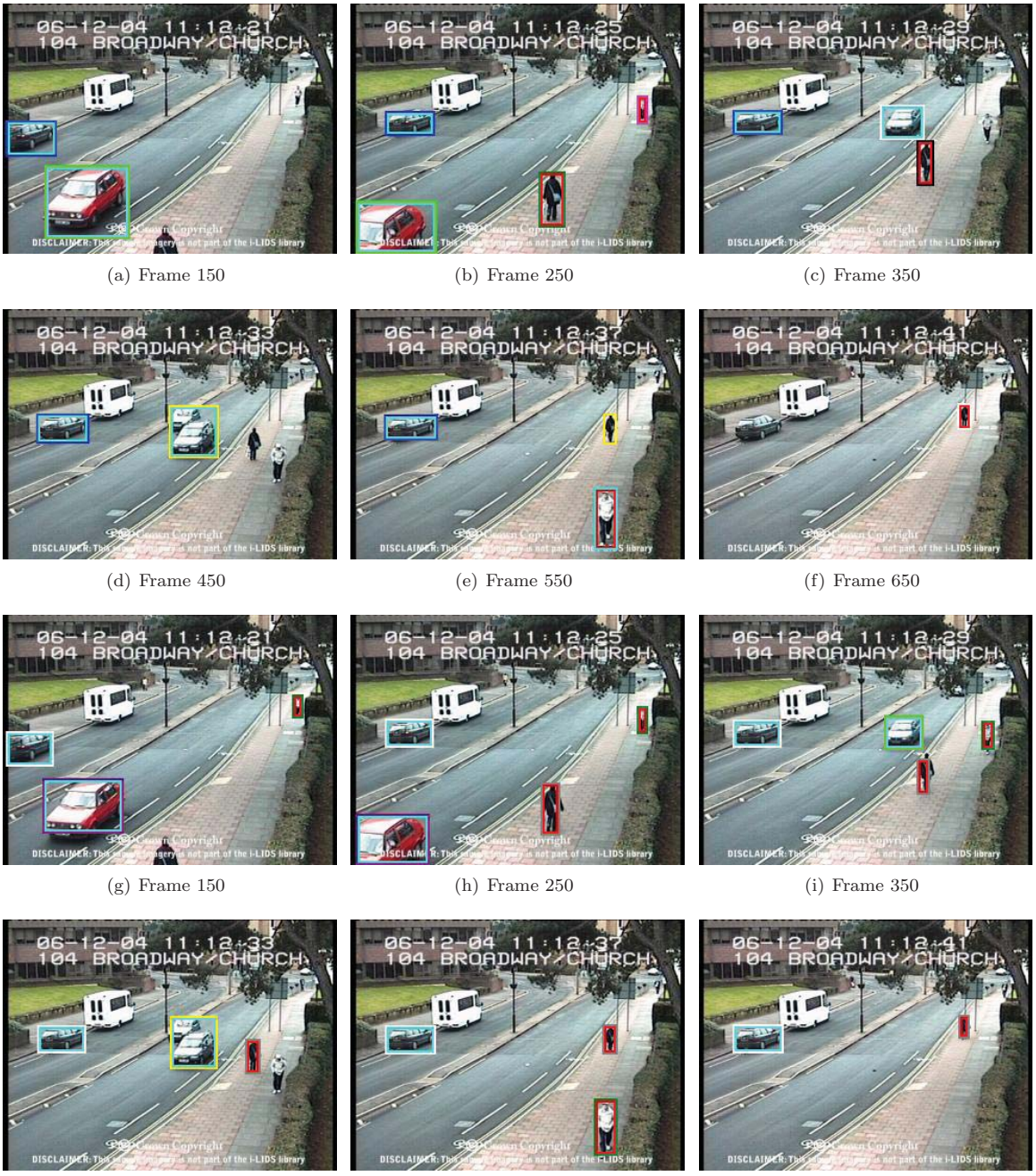
[15] Silogic and Inria. Etiseo metrics definition (http://www-sop.inria.fr/orion/etiseo/download.htm). Technical report, 6th January 2006.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, page 252 Vol. 2, 1999.

[17] T. Yamane, Y. Shirai, and J. Miura. Person tracking by integrating optical flow and uniform brightness regions. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3267–3272 vol.4, 1998.

[18] M.-T. Yang, Y.-C. Shih, and S.-C. Wang. People tracking by integrating multiple features. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, pages 929– 932, 2004.

[19] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1208–1221, 2004.



(a) Frame 150 (b) Frame 250 (c) Frame 350

(d) Frame 450 (e) Frame 550 (f) Frame 650

(g) Frame 150 (h) Frame 250 (i) Frame 350

(j) Frame 450 (k) Frame 550 (l) Frame 650

Figure 7. Example output from RD7 - Maintaining Tracking of Temporarily Stopped Objects.

(a) Frame 2100 (b) Frame 2150 (c) Frame 2200

(d) Frame 2350 (e) Frame 2375 (f) Frame 2400

(g) Frame 2100 (h) Frame 2150 (i) Frame 2200

(j) Frame 2350 (k) Frame 2375 (l) Frame 2400

Figure 8. Example output from RD7 - Improved Detection and Localisation of Objects.